CSC 450-750 Assignment 3

Due date: Nov 21th, 2021 by 11:59 PM

Learning Objectives:

- Create a protocol based on different entities in the scenario, and deploy it to AWS.
- Correctly set up agents to satisfy the requirements.
- Ensure the safety and liveness across the whole protocol.
- Learning how enactment can be placed inside of protocol.

Scenario:

Consider each entity as an agent in the protocol. Unlike the previous two assignments, we are only considering three agents: Buyer, Merchant, and Shipping service. Please note that not all the following needs to be transferred into schemas, it is there to help you better understand where and when it should happen.

- Buyer:
    - Request quote from merchant
    - Place an order with shipping address
    - Submit payment to merchant
    - Send cancel order request to merchant only before item has been shipped (only available when item has not been shipped, otherwise it is failed)
    - Receive item from shipping service, print out date that received
    - Receive refund from merchant for canceled order
    - Receive refund from merchant for defective item
- Merchant:
    - Send out quote after getting buyer's request
    - Receive cancellation request from buyer, refund the buyer, and take no further action since then
    - Send shipped notification to buyer if:
        1. Order is not canceled
        2. Received order details with buyer's information
        3. Received payment
    - Send the item to the shipper for shipment with buyer's address and name
    - Send defective notification to buyer if item is defective, refund buyer
    - Remind the buyer that item has been delivered with delivery confirmation from Shipping service to buyer
- Shipping service
    - Receive buyer information from merchant
    - Pack and ship the item to the buyer
    - Provide tracking number to merchant

    - Report defective item to merchant if item is defective on its way to buyer
    - Send confirm delivery message to merchant

Setup Hints:

- You will need an AWS account, please go to the following link to set up your I AM user: https://www.serverless.com/framework/docs/providers/aws/guide/credentials
- Please make sure to link a card with the aws account, we are using the free service so AWS will not charge you for using their service (they may charge for $1 for checking if the card is valid), but without this information you will not be able to use their service.
- Please comment out the framework version in serverless.yml in adapter folder, otherwise you will have a version error.
- When set up your serverless with AWS key, please remove the <> in the instruction to avoid “\n” error

Deliverables:

- A protocol in BSPL with roles corresponding to the above stakeholders and messages through which they can accomplish the specified interactions
    - The protocol would have a parameters line that expresses what information is needed to complete the transaction.
    - The protocol should be safe and live, as verified using the Protocheck tool.
- Sample agents, where each agent adopts a role in the protocol. Call these agents Buyer-A, Merchant-A, Shipping-A.
    - These agents send and receive messages according to their role in the protocol. The agents should be designed so that the transaction completes.
    - You can find the Deserv library (similar to Bungie, but for implementing AWS agents) along with an example here: https://gitlab.com/masr/deserv/-/tree/master/serverless
- Two sample enactments of the protocol as generated by the above agents interoperating (i.e., working together).
    - An enactment equals a list (also called vector) of observation sequences (also called history), one sequence or history per agent.
    - Two enactments can differ in the order of observations made by at least one agent
- An alternative agent for the Buyer role. Call this agent Buyer-B.
    - Buyer-B should interoperate with the remaining agents, i.e., Merchant-A, Shipping-A.
    - One enactment involving Buyer-B that is different from any enactments generated by Buyer-A

Evaluation:

- Learn and deploy the agent to AWS
- Provide at least two enactments in your protocol
- Each entity transmits a corresponding message based on its activity. For example, the buyer provides the item's name as a message of placing an order.
- Your protocol will be examined based both on what you write and how the participating agents enact it.
- Your agents will be examined based on their ability to generate all legal enactments.

Grading:

1. One python file for each agent, one protocol bspl file, one configuration file, with one serverless yml file that can be used to recreate the protocols and successfully handle a transaction. (90 points)
2. A pdf file with the explanation of your protocols. (10 points)

Submission:

Please make sure to put all of your source files and a PDF version of your README file in one folder, and store it inside of a zip file. The zip file should include your name and the abbreviation of assignment, e.g. Ezio-Mei-P3.zip. Submit this zip file by the due date on moodle.

Please use the discussion forum on Moodle for this assignment to talk about general concerns and post questions about the tool and so forth. Please do not post any part of your answers publicly; instead, send questions specific to your approach via email. When you write to Ezio, cc Professor Singh so we can try to help where possible. (As usual, we will try not to provide you solutions but will guide you where we can).